**Текст к презентации «Дерево секущих гиперплоскостей»**

**Слайд 1**

Приветствие.

**Слайд 2**

План презентации. Я расскажу про основные недостатки дерева решений, а также про предлагаемый нами метод машинного обучения – дерево секущих гиперплоскостей.

**Слайд 3**

Дерево решений — метод представления решающих правил в определенной иерархии, включающей в себя элементы двух типов — узлов и листьев. Узлы включают в себя решающие правила и производят проверку примеров на соответствие выбранного атрибута обучающего множества.

Основные плюсы дерева:

1. Быстрая скорость обучения

2. Показали практическую применимость при использовании бустинга.

Основные минусы:

1. Не учитываются зависимости между признаками
2. Не решает задачу

**Слайд 4**

Например, для случая, приведенного на слайде, более корректным выглядит построение линейной разделяющей гиперплоскости – логистической регрессии.

**Слайд 5**

Таким образом, мы приходим к идее дерева секущих гиперплоскостей, в котором в каждом узле принятие решений будет осуществляться исходя из того, находится наблюдение выше или ниже гиперплоскости.

В целом дерево секущих гиперплоскостей можно представить в виде дерева решений, представленном на слайде. Как видно из рисунка, в каждом узле логическое правило формируется гиперплоскостью.

**Слайды 6 - 10**

Решение о классификации в дереве секущих плоскостей принимается последовательно, первый узел дерева отсекает часть наблюдений и принимает решение об их классификации, от классификации остальных наблюдений он отказывается. Затем во втором узле дерева отсекается часть оставшихся наблюдений и принимается решение об их классификации и так далее.

На слайде представлен пример классификации деревом секущих гиперплоскостей.

**Слайд 11**

Итого 100% разделения данного множества на обучающей выборке потребовалось 3 линейных классификатора.

**Слайд 12**

Перейдем к рассмотрению разработанной модели

**Слайд 13**

Для построения члена комитета, отсекающего класс 1 целевая функция имеет вид, представленный на слайде.

L - некоторое число, влияющее на долю класса 0 в отсекающей гиперплоскости (при больших L член комитета будет полностью отказываться от классификации, при малых L доля класса 0 будет существенной. Далее будет описан алгоритм подбора L).

Функция принимает значения 0 для всех случаев, когда член комитета отказывается от классификации (точки лежат ниже гиперплоскости - значения ). В ином случае, для класса 0 функция принимает следующий вид ), для класса 1 функция принимает вид -). Соответственно, минимизируя данную функцию, мы будем стремиться к увеличению числа точек класса 1, лежащих выше гиперплоскости, при одновременном уменьшении числа точек класса 0.

Узел дерева может голосовать за принадлежность к классу 1 или классу 0. Очевидно, что в случае, если требуется построить узел, голосующий за класс 0, то необходимо заменить на (1-.

**Слайд 14**

Для построения дерева секущих гиперплоскостей с использованием функции ReLU1 предлагается следующий жадный алгоритм.

В качестве гиперпараметра алгоритма требуется задать число наблюдений (N), находящихся выше гиперплоскости, данный параметр схож с минимальным числом наблюдений на листе у дерева.

**Шаг 1**. Осуществляется построение 2 гиперплоскостей (голосование за класс 1 / класс 0), при этом L определяется как 2^k (k=10);

**Шаг 2**. Из 2 гиперплоскостей отбирается гиперплоскость с наибольшим числом наблюдений, лежащих выше гиперплоскости (голосование). В случае, если число наблюдений меньше N, выполняется шаг 1 с k=k-1, и так далее, до нахождения k, при котором число наблюдений выше гиперплоскости будет больше N;

**Шаг 3**. Наблюдения, которые находятся выше гиперплоскости, исключаются из выборки, вероятности принадлежности к классу определяются исходя из долей классов. Далее возвращаемся к шагу 1.

Построение новых членов комитета продолжается, пока количество наблюдений для классификации превышает 2\*N.

**Слайд 15**

В результате работы алгоритма на открытых датасетах UCI репозитория были получены следующие результаты.

**Слайд 16**

В целом, в большинстве случаев дерево секущих имеет качество выше чем у дерева решений, но при этом проигрывает ансамблям на деревьях (Random forest, LightGBM). Заметим, что дерево секущих гиперплоскостей также может использоваться в бустингах, и можно ожидать получение большей точности по сравнению с бустингами классического дерева.

**Слайд 17**

Для исследования склонности к переобучению, дерево решений и дерево секущих гиперплоскостей были обучены на датасете Устойчивость эл. сети. На слайде приведены зависимости метрики GINI на обучающей и тестовой выборках от глубины решающего дерева и числа наблюдений N, находящихся выше гиперплоскости, дерева секущих плоскостей соответственно.

Из графиков видно, что у дерева секущих гиперплоскостей имеется локальный оптимум, когда переобучение алгоритма минимально, а также качество на тестовой выборке достигает приемлемого значения.

Заметим, что при увеличении числа N качество дерева секущих гиперплоскостей падает не так сильно, как качество решающего дерева при увеличении аналогичного параметра. Это может говорить о большей стабильности метода при использовании его в качестве базового алгоритма в ансамблях деревьев (бустингах).

**Слайд 18**

Спасибо за внимание! Я готов ответить на ваши вопросы